



EDB

Migration Portal

Release 4.0.0

EDB

May 02, 2022

CONTENTS

1	Release notes	3
1.1	Version 4.0	3
2	Supported platforms	5
2.1	Supported browsers	5
3	Using Migration Portal	7
3.1	Overview of the Migration Portal home page	8
3.2	Overview of the Migration Portal Projects page	9
3.3	Migration Portal quick start To migrate Oracle schemas using Migration Portal:	10
4	Migrating a database	13
4.1	Performing a schema extraction	13
4.1.1	Extracting schemas using the EDB DDL Extractor	13
4.1.2	Schemas and objects support	16
4.1.3	Wrapped objects	17
4.1.4	About file encoding	17
4.2	Performing a schema assessment	19
4.2.1	Perform a schema assessment	19
4.2.2	Generate an assessment report	22
4.3	Schema migration	23
4.3.1	Migrating to an existing on-premises EDB Postgres Advanced Server host	24
4.3.2	Migrating to a new on-premises EDB Postgres Advanced Server installation	25
4.3.3	Migrate to the cloud	26
4.4	Data migration	26
5	Advanced data migration	29

Migration Portal is a web-based service for migrating Oracle database schemas to the EDB Postgres Advanced Server platform. The Migration Portal assesses and analyzes Oracle database schemas and converts:- Types- Tables- Sequences- Constraints- Triggers- Views- Stored procedures- Packages- Database links (dblinks)- Materialized views- Indexes

The migration produces DDLs compatible with EDB Postgres Advanced Server.

The user-friendly portal interface simplifies the assessment and migration process. Log in to the portal and start the migration process .

EnterpriseDB has helped companies migrate their existing database systems to Postgres for years. For more information, see the

[EnterpriseDB website](#) .

RELEASE NOTES

The Migration Portal documentation describes the latest version of Migration Portal 4.0.

Version	Release date
[4.0](10_mp_4.0_rel_notes)	22 Apr 2022

1.1 Version 4.0

New features, enhancements, bug fixes, and other changes in Migration Portal 4.0 include the following:

Type	Description
Enhancement	The Migration Portal now supports the use of the EDB DDL Extractor again. Issues related to the DDL Extractor are resolved and the temporary restriction introduced on the use of DDL Extractor file format has been removed. The Migration Portal continues to support the use of the Oracle Data Pump utilities following the instructions in the [documentation](https://www.enterprisedb.com/docs/migration_portal/latest/04_mp_migrating_database/01_mp_schema_extraction/); however, the use of the EDB DDL Extractor is recommended.
Enhancement	The schema assessment algorithms are optimized to address a few performance related issues and to reduce overall schema assessment time.
Enhancement	Added a repair handler to remove CACHE/NOCACHE keyword from TABLE definition as EDB Postgres Advanced Server does not support it.
Security fix	The jackson-databind and h2 libraries are upgraded to the latest versions to address security issues (CVE-2020-36518, CVE-2022-23221, CVE-2021-42392 and CVE-2021-23463).
Bug fix	Fixed the grammar rule to resolve a parsing error that skipped execution of a repair handler to remove EDITIONABLE keywords.
Bug fix	Fixed the repair handler to perform proper conversion of Oracle PIPELINED and PIPE ROW function calls.
Bug fix	Fixed an issue where the case sensitivity was lost when an object name was based on Unicode characters.
Bug fix	Fixed the issue that displays the Getting Started wizard when Migration Portal landing page is opened for the first time for a new user.
Bug fix	Fixed the issue where a VIEW definition was truncated during transformation in the target EPAS DDL script.

continues on next page

Table 2 – continued from previous page

Type	Description
Bug fix	Previously, uploaded DDL files having characters that were not compatible with UTF-8 were failing at the start of the Assessment phase. Migration Portal now reads the SQL files as UTF-8. In case a non UTF-8 file is uploaded, all the characters that are not compatible with UTF-8 will be converted to the replacement character ‘ ’ and assessment will continue without failing.
Bug fix	Fixed constraint name assignment for the unnamed constraint in the source Oracle DDL.
Bug fix	Updated the repair handler to remove the OIDINDEX storage clause for Oracle Data Pump generated source DDL files.
Bug fix	Allowed the COLLATE and COMPUTATION keywords to be used as identifiers.
Bug fix	Fixed the handling of multi-line comments that appear as part of single line comments.
Bug fix	Fixed the grammar to properly identify and skip wrapped objects.
Bug fix	Fixed ConcurrentModificationException occurred while running multiple projects simultaneously.
Bug fix	Fixed unique object names handling across multiple schemas. The object names are not required to be unique across different schemas.
Bug fix	Made some fixes in the create table grammar to ignore the order of some clauses as no specific order is required.

SUPPORTED PLATFORMS

The Migration Portal supports assessment and migration from Oracle 11g, 12c, 18c, and 19c to EDB Postgres Advanced Server 10, 11, 12, 13, or 14. Migration Portal is supported on the following browsers.

2.1 Supported browsers

For the best user experience, we recommend using the Google Chrome browser. Migration Portal is also supported on the following browsers:

Browser	Supported version
Apple Safari on Macintosh OS	14 and above
Google Chrome	68 and above
Microsoft Edge	42 and above
Mozilla Firefox	60 and above

USING MIGRATION PORTAL

Migration Portal allows you to easily migrate your database schema from Oracle to EDB Postgres Advanced Server. You can upload schemas for assessment and get immediate feedback and suggestions. The portal allows you to download assessed DDLs for all objects and create your EDB Postgres database on premises or in the cloud.

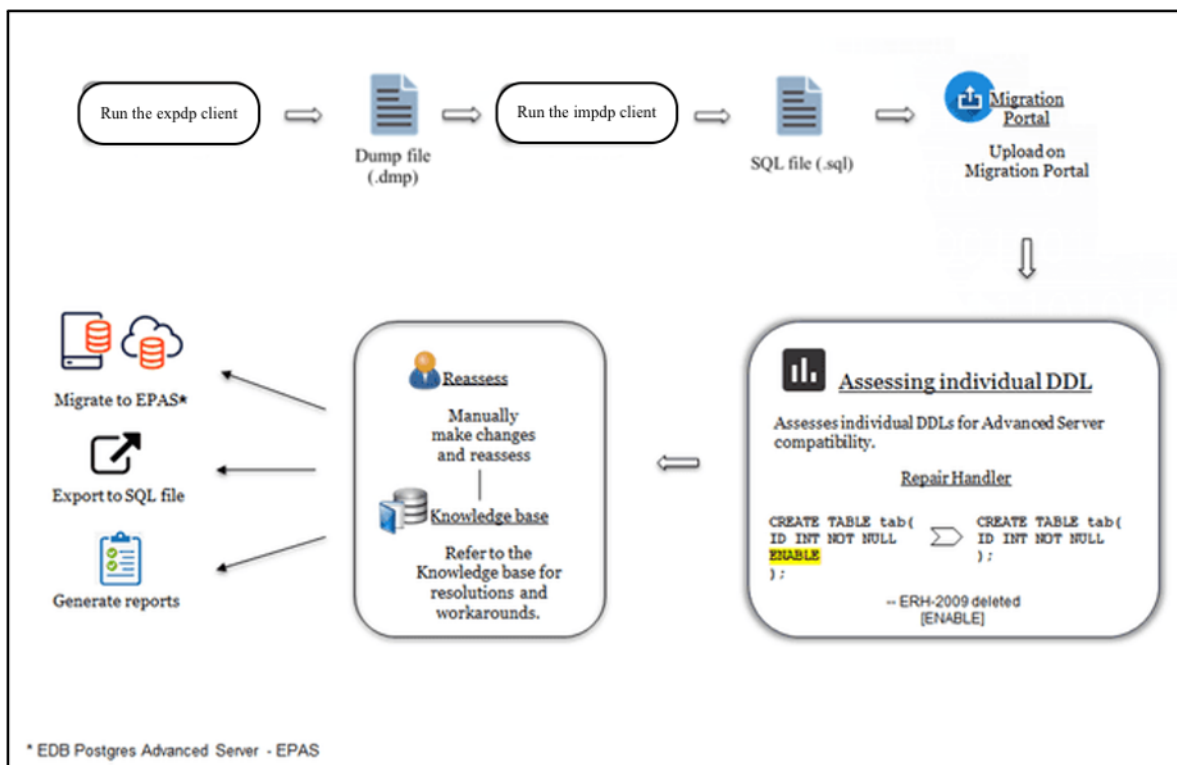


Fig. 1: The assessment and migration process

Note: Migration Portal currently doesn't have team collaboration features. We recommend collaborative meetings, screen sharing, and using the downloadable assessment report to share results.

To access the Migration Portal:

1. Open a browser and navigate to <https://www.enterprisedb.com/>.
2. On the EnterpriseDB home page, select **PostgreSQLSoftware>Migration**.
3. On the Migration Portal page, select **Findoutmore>OpenMigrationPortal**.

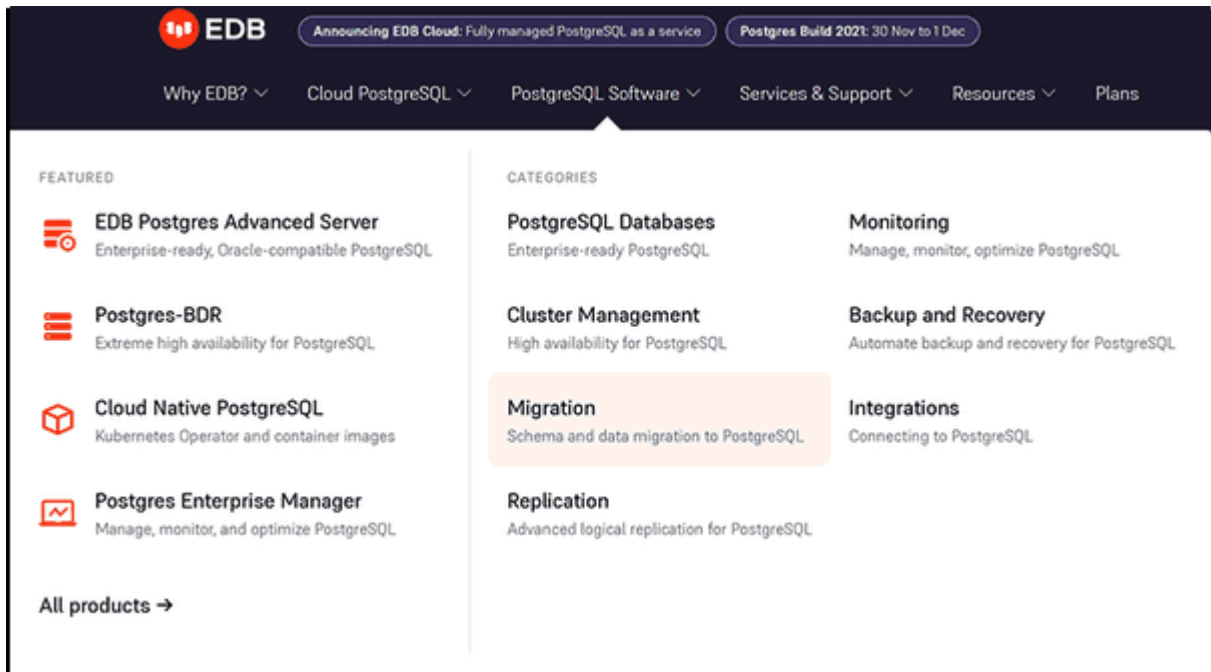


Fig. 2: Migration Portal on the EnterpriseDB home page

4. Log in using your credentials. See [Migration Portal quick start To migrate Oracle schemas using Migration Portal](#): for an overview of using Migration Portal to migrate Oracle schemas.

3.1 Overview of the Migration Portal home page

The Migration Portal home page provides quick access to migration tools.

The Migration Portal home page allows access to the following Migration Portal features:

- **Projects** : The Projects panel displays a list of assessed projects.
- **Create project** : Select **+New** to create a project.
- **Search projects** : Use the **Search** box to search for projects.
- **Overview** : The Overview panel provides details about the selected project and displays the compatibility percentage after schema assessment.
- **Migrate to** : Select **Migrate to** either to download an EDB Postgres Advanced Server compatible .sql file or migrate a schema to an EDB Postgres Advanced Server on premises or in the cloud. For more information, see [Schema migration](#) .
- **Report** : Select **Report** to view and download the schema assessment report.
- **Delete** : Select **Delete** to delete a selected project.
- **Warningsign** : A warning message appears if a project or a schema is less than 70% compatible or any DDL doesn't succeed after multiple attempts.
- **UploadDDLfile** : Select **Uploadnewschema** to upload a new DDL file.
- **Schemas** : The Schemas panel displays the assessment result from an uploaded DDL file.
- **Quickhelp** : The Quick help panel contains quick links to a few required resources.

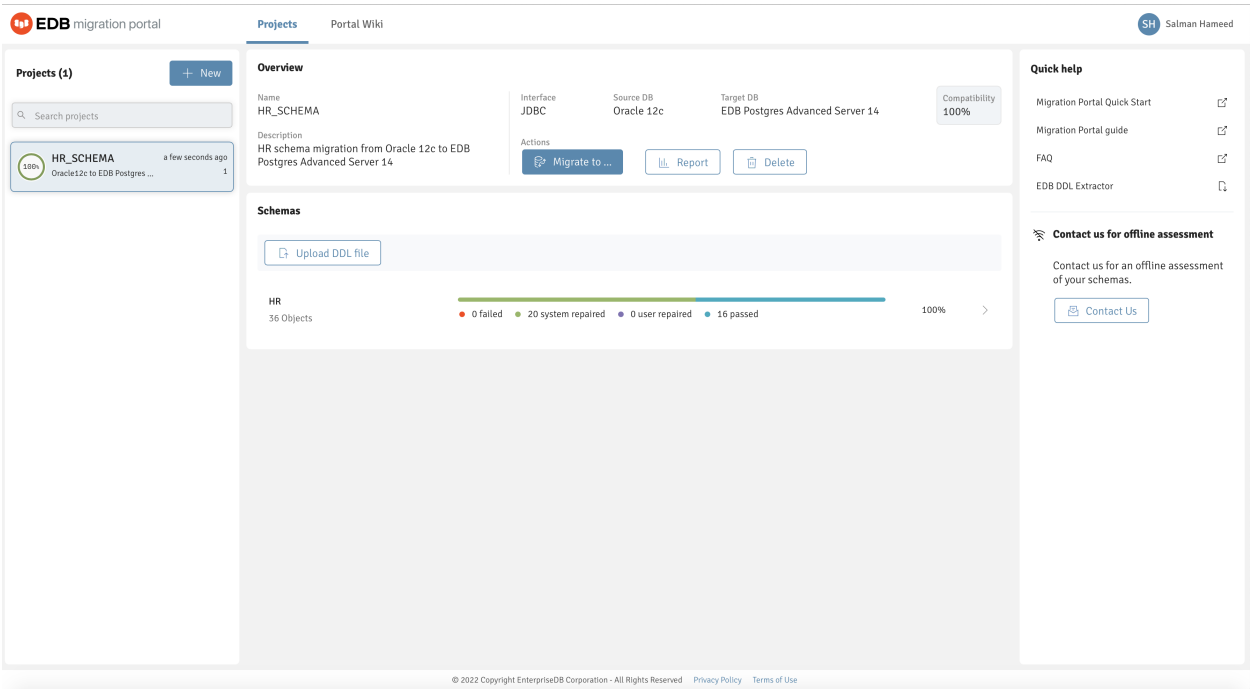


Fig. 3: The Migration Portal home page.

- **PortalWiki** : Select **PortalWiki** to access links to product information and more help guides.

3.2 Overview of the Migration Portal Projects page

The Migration Portal Projects page provides detailed information about your migration.

Use the following resources to gather information about your migration projects:

- **Compatible** : The **Compatible** gauge displays the color on the basis of the compatibility percentage of the assessed schema.
- **SchemaCount** : Displays the number of schemas in a project.
- **Migrateto** : Select **Migrateto** to migrate the schema to an EDB Postgres Advanced Server on premises or in the cloud.
- **Searchobjects** : Use the **Search** box to search for objects.
- **Filters** : From the left panel of the Projects page, you can filter the system repaired and manual repaired objects. You can select one or more filter combinations to refine the information.
- **Objects** : Displays the objects for the selected schemas.
- **CommonFailures** : Displays the reason for the failed objects for the selected schemas.

Note: You can download a `.csv` file for the common failures for the project.

- **Schema** : Displays the result of the assessment.
- **Tooltip** : Hover over a result set to display a tooltip with the number of passed, failed, and repaired objects.

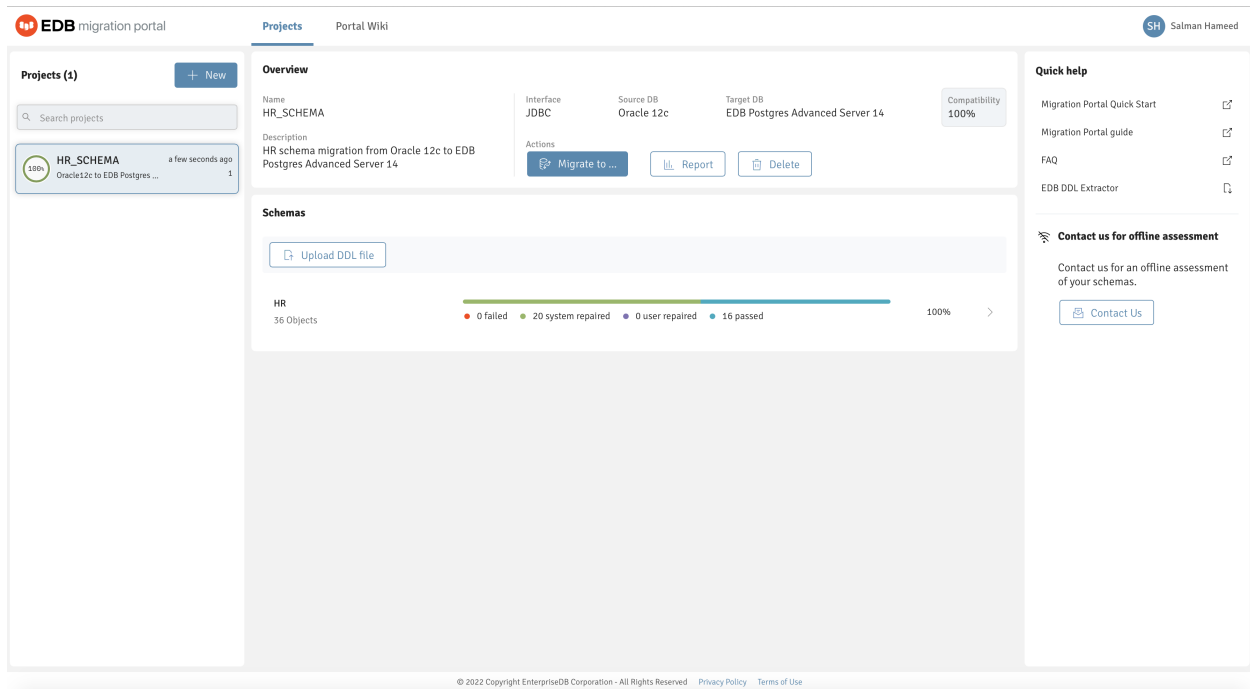


Fig. 4: The Migration Portal Projects Page overview.

- **Quickhelp** : The Quick help panel displays links to Knowledge Base articles and repair handler documentation.
- **Search** : Use the **Search** box to search the Knowledge Base entries or repair handler documentation for specific information.

3.3 Migration Portal quick start To migrate Oracle schemas using Migration Portal:

1. Generate a .sql file with metadata information of your schemas. For more information, see *Performing a schema extraction* .
2. Log in to *Using Migration Portal* with your EDB credentials.
3. Create a new project.
4. Upload the .sql file with your schema metadata.

Note: Uploading large DDL files might freeze or crash the browser (depending on the browser and the system resources in use). In this scenario, try running Migration Portal in a different browser or extracting fewer schemas and uploading that file into Migration Portal.

1. Select **Create&assess** . Migration Portal reviews every construct, executes repair actions to improve compatibility with EDB Postgres Advanced Server, and flags any remaining errors that require manual intervention.
2. Verify the DDL objects, for example, TABLES that don't show a 100% success ratio.
3. Select the objects that aren't compatible with EDB Postgres Advanced Server.

4. Refer to the Knowledge Base for checking the possible workarounds for the objects that aren't compatible in EDB Postgres Advanced Server.
5. Make the required changes and select **Reassess** to rerun the analysis.
6. After resolving errors in your schemas, you can download the modified EDB compatible DDL file and use any client application such as pgAdmin or PSQL client and migrate the schema to EDB Postgres Advanced Server. You can also migrate to EDB Postgres Advanced Server on the cloud. For more information, see [Schema migration](#) .

MIGRATING A DATABASE

To migrate a database, you must complete the following steps:

1. Perform a *Performing a schema extraction* .
2. Perform a *Performing a schema assessment* .
3. Perform a *Schema migration* .
4. Perform a *Data migration* .

4.1 Performing a schema extraction

You can perform a schema extraction using either of the following methods. EDB recommends using the EDB DDL extractor to extract your schemas.

- *Extracting schemas using the EDB DDL Extractor* (recommended method)
- Oracle's Data Pump utility

4.1.1 Extracting schemas using the EDB DDL Extractor

Download the latest EDB DDL Extractor script from the Migration Portal Projects page.

Note: Migration Portal might become unresponsive for very large SQL files, depending on your system and browser resource availability. To resolve this, try extracting fewer schemas at a time to reduce the SQL file size.

Prerequisites

You can run the EDB DDL Extractor script in SQL Developer or SQL*Plus. It uses Oracle's DBMS_METADATA built-in package to extract DDLs for different objects under schemas (specified while running the script). The EDB DDL extractor creates the DDL file uploaded to the portal and analyzed for EDB Postgres Advanced Server compatibility.

Note: You must have *CONNECT* and *SELECT_CATALOG_ROLE* roles and *CREATE TABLE* privilege.

For SQL*Plus

1. Connect to SQL*Plus and run this command:

```
SQL>@edb_ddl_extractor.sql
```

2. When prompted, if the Oracle user that is running the script has the required privileges, press **Enter** to continue running the script.
3. When prompted, provide the schema name. When extracting multiple schemas, use a comma (' , ') as a delimiter.

Note: If you want to extract all the user schemas from the current database, don't mention any schema names while extracting. However, we recommend mentioning the schema names that you want to extract.

4. When prompted, enter the path or directory for the extractor to store the extracted DDL. If you don't specify a path, the extracted DDL is output to the directory where you ran the SQL*Plus command.
5. When prompted, enter yes or no depending on whether you want to extract dependent objects from other schemas.

For example, on Linux:

```
Enter a comma separated list of schemas to be extracted (Default all schemas): HR, SCOTT,
↪ FINANCE

Location for output file (Default current location) : /home/oracle/extracted_ddls/

WARNING:

Given schema(s) list may contain objects which are dependent on objects from other
↪ schema(s), not mentioned in the list.` `Assessment may fail for such objects. It is
↪ suggested to extract all dependent objects together.

Extract dependent object from other schemas?(yes/no) (Default no / Ignored for all
↪ schemas option): yes
``` On Windows:

```text
Enter comma separated list of schemas to be extracted (Default all schemas): HR, SCOTT,
↪ FINANCE

Location for output file (Default current location) : c:\Users\Example\Desktop\

WARNING:

Given schema(s) list may contain objects which are dependent on objects from other
↪ schema(s), not mentioned in the list.` `Assessment may fail for such objects. It is
↪ suggested to extract all dependent objects together.

Extract dependent object from other schemas?(yes/no) (Default no / Ignored for all
↪ schemas option): yes
```

For SQL Developer

After loading the `edb_ddl_extractor.sql` script into SQL Developer and connecting to the source Oracle database, run the script. As the script executes, respond to the prompts.

1. If the the user for the database connection has the required privileges as listed in the **ScriptOutput** tab, select **Yes** to continue running the script.
2. Enter a comma-separated list of schemas and select **OK**.
3. Enter the path for the output file and select **OK**. The default is the location of the DDL Extractor script.
4. Enter yes or no to extract dependent objects and select **OK**. The default is to not extract dependent objects.

Output of the DDL Extractor run appears in the Script Output tab. The name of the output file appears after the **Extraction Completed** message in the script output.

Note: You can also enter single schema name in both SQL*Plus and SQL Developer.

The script then iterates through the object types in the source database. Once the task is completed, the .SQL output is stored at the location you entered (e.g., `c:\raw-latex:\Users\Example\raw-latex\DesktopThe`script` then iterates through the object types in the source database.).

See an important [Release notes](#) about the file encoding format expected by Migration Portal.

Additional notes

- The EDB DDL Extractor script doesn't extract objects restored using Flashback that still have names like `BIN$b54+4XlEYwPgUAB/AQBWwA==$0`. If you want to extract these objects, you must change the names of the objects and rerun the extraction process.
- The EDB DDL Extractor extracts `nologging` tables as normal tables. Once these tables are migrated to EDB Postgres Advanced Server, WAL log files are created.
- The EDB DDL Extractor extracts objects only with `VALID` status. For any objects that have `INVALID` status that you want Migration Portal to assess, first update them to `VALID`.
- The EDB DDL Extractor doesn't extract objects that were obfuscated using the Oracle wrap feature. As such, these objects aren't included in the set of DDL assessed by the Migration Portal. If you want to assess these objects and migrate them to EDB Postgres Advanced Server, replace the wrapped versions of the objects with nonwrapped versions. See [Wrapped objects](#) for more information.
- The EDB DDL Extractor creates Global Temporary tables to store the schema names and their dependency information. These tables are dropped at the end of successful extraction.
- The EDB DDL Extractor script doesn't extract schemas whose name starts with `PG_` because PostgreSQL doesn't support it. If you want to extract these schemas, you must change the name of the schema before extraction.

4.1.2 Schemas and objects support

The lists and tables that follow show supported and unsupported schemas and objects.

Unsupported schemas

Exclude these Oracle systems schemas while generating the SQL dump file.

ANONYMOUS	APEX_PUBLIC_USER	APEX_030200	APEX_040000	APEX_040200
APPQOSSYS	AUDSYS	CTXSYS	DMSYS	DBSNMP
DBSFUSER	DEMO	DIP	DMSYS	DVF
DVSY	EXFSYS	FLWS_FILES	FLWS_020100	FRANCK
GGSYS	GSMAD-MIN_INTERNAL	GSMCATUSER	GSMROOTUSER	GSMUSER
LBACSYS	MDDATA	MDSYS	MGMT_VIEW	OJVMSYS
OLAPSYS	ORDPLUGINS	ORDSYS	ORDDATA	OUTLN
ORACLE_OCM	OWBSYS	OWBYSS_AUDIT	PDBADMIN	RMAN
RE-MOTE_SCHEDULER_AGENT	SI_INFORMTN_SCHEMA	SI_TIAL_CSW_ADMIN_USR	SPA-URL_WFS_ADMIN_USR	SQLTXADMIN
SQLTXPLAIN	SY\$UMF	SYS	SYSBACKUP	SYSDG
YSKM	YSRAC	SYSTEM	SYSMAN	TSMSYS
WKPROXY	WKSYS	WK_TEST	WMSYS	XDB
XS\$NULL				

Note: EDB Postgres Advanced Server doesn't support schema names starting with *pg_* .

Supported object types

- Synonyms
- DB links
- Types and type body
- Sequences
- Tables
- Constraints
- Indexes (Except LOB indexes and indexes on materialized views)
- Views
- Materialized views
- Triggers
- Functions
- Procedures
- Packages

Note: COMMENTS on columns, tables, and materialized views are also supported.

Unsupported object types

- Editions
- Operators
- Schedulers
- LOB indexes and indexes on materialized views
- XML schemas
- Profiles
- Role and object grants
- Tablespaces
- Directories
- Users
- RLS policy
- Queues

4.1.3 Wrapped objects

Migration Portal can't assess wrapped objects. If you include them in the extracted DDL, they aren't loaded into Migration Portal and aren't included in the count of objects that are assessed. If you want to assess wrapped objects and migrate them to EDB Postgres Advanced Server, include unwrapped versions of the objects in the DDL file that you upload to Migration Portal. The recommended way of doing this is to replace the wrapped versions in the Oracle database with clear-text versions before performing the schema extraction. After performing the schema extraction, you can replace the objects with the wrapped versions.

4.1.4 About file encoding

Migration Portal recommends the .SQL output file be in the UTF-8 encoding format. If you upload a .SQL file with non-UTF-8 encoding, all the characters that aren't compatible with UTF-8 are converted to the replacement character ” in the output DDL.

Tip: You can manually convert the extracted file to the UTF-8 format by using the iconv utility on Linux or the LibIconv utility on Windows. For example, if your database character set is in Latin-1 (ISO-8859-1), you can convert the extracted file to the UTF-8 format, as follows:

```
iconv -f iso-8859-1 -t UTF-8 sample.sql > sample_utf8.sql
```

Extracting schemas using data pump utility

Migration Portal requires only the metadata in the SQL dump file to assess the extracted schemas.

Prerequisites

- If you plan on exporting schemas that aren't your own, make sure that you're assigned the DATAPUMP_IMP_FULL_DATABASE role. Otherwise, you can export only your own schema.
- Make sure that you have enough tablespace quota to create objects in the tablespace.

```
SQL> create directory DMPDIR as '/sample/file/path'; OnWindows :
```

```
SQL> create directory DMPDIR as 'c:\sample\file\path\';
```

- In SQL*Plus, grant read-write permissions to the user running the Data Pump utility.

```
SQL> grant read,write on directory DMPDIR to <Username>;
```

Perform either of the following procedures:

- *Extract one or more schemas in a database*
- *Extract all schemas in a database*

Extract one or more schemas in a database

1. Before running the `expdp` command, create a file with a `.par` extension (for example, `export.par`) on your server. Add attributes and values to the file:

edb_notranlate_0 The attributes and values in this command specify the following options: -
CONTENT=metadata_only extracts only the metadata of the schemas. - INCLUDE= specifies the database object types to include in the extraction. Append a comma-separated list of the

Supported object types to extract only database object types that are supported

by Migration Portal.

2. From the command line, run the export command to generate a `schemas_metadata.dump` file. For example, to extract metadata information for <Schema_1>, <Schema_2>, <Schema_3>, and so on, run: edb_notranlate_1
3. To generate a SQL file from the dump file, run the import command.

For example, to generate a `YourSchemas.SQL` file from the `schemas_metadata.dump` file, enter: edb_notranlate_2
See an important

Release notes about the file encoding format expected by the Migration

Portal.

Extract all schemas in a database

Note: Don't perform this procedure from a user account that belongs to the excluded schemas list (see *Unsupported schemas*). The *impdp* command fails if the user account running the command is in the excluded list of schemas.

1. Before running the *expdp* command, create a file with a *.par* extension (for example, *export.par*) on your server. Add attributes and values to the file:

edb_notranlate_3 The attributes and values in this command specify the following options: - *CONTENT=metadata_only* extracts only the metadata of the schemas. - *INCLUDE=* specifies the database object types to include in the extraction. Append a comma-separated list of the

Supported object types to extract only database object types that are supported

by Migration Portal. - *FULL=Y* extracts all schemas in the database.

2. From the command line, run the *export* command to generate a *schemas_metadata.dump* file: edb_notranlate_4
3. Before running the *impdp* command, create a parameter file with a *.par* extension (for example, *import.par*) on your server. Add attributes and values to the file: edb_notranlate_5 The attributes and values in this command specify the following options: - *EXCLUDE* specifies the schemas to exclude. Append a list of the

Unsupported schemas to ensure that the Data Pump utility doesn't extract any system schemas (including schemas beginning with *pg_*).

4. To generate a SQL file from the dump file, run the *import* command.

For example, to generate *YourSchemas.SQL* file from the *schemas_metadata.dump* file, enter: edb_notranlate_6 See an important

Release notes about the file encoding format expected by the Migration

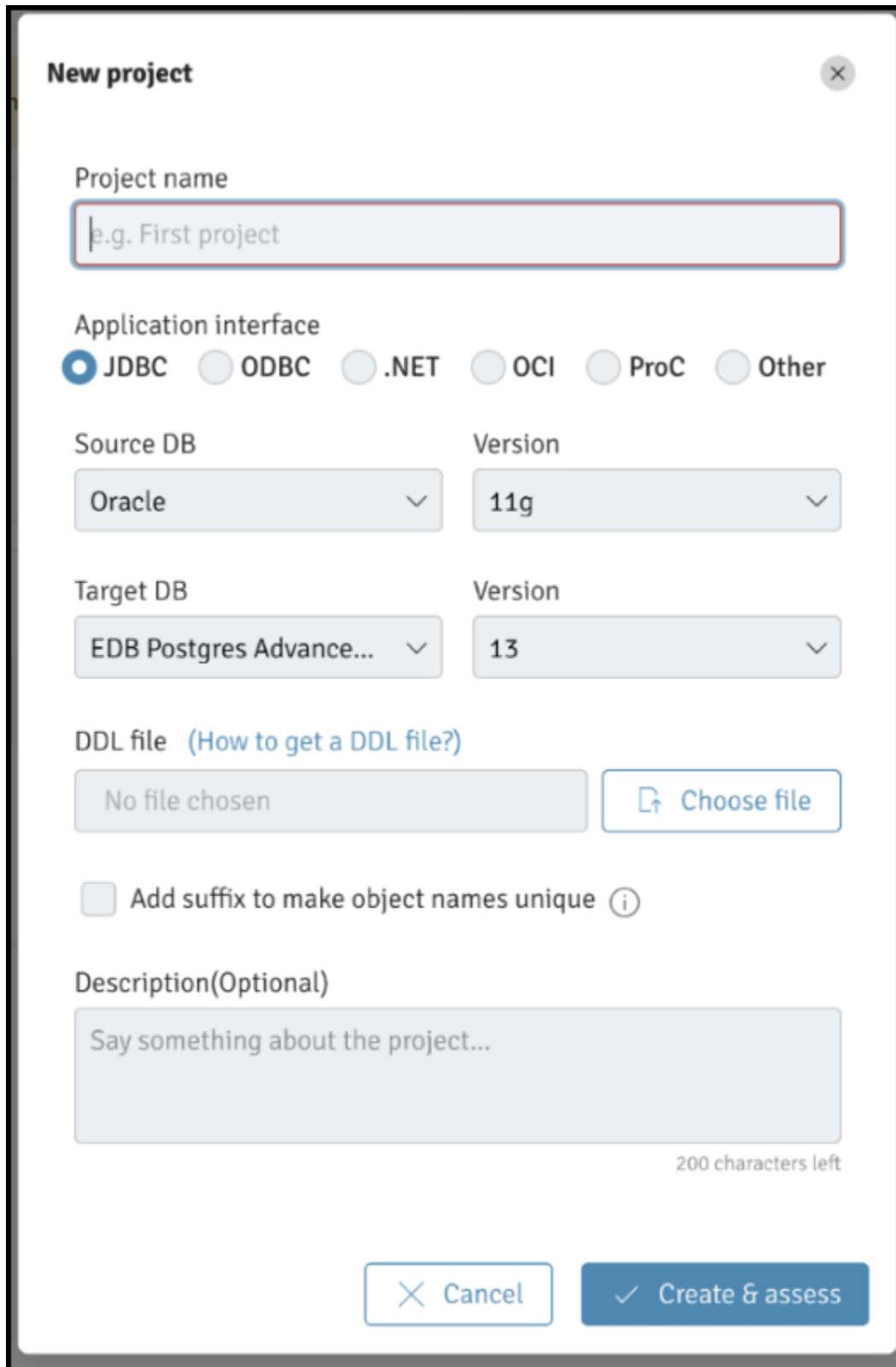
Portal.

4.2 Performing a schema assessment

You can assess an Oracle database schema for compatibility with EDB Postgres Advanced Server. You can also generate a report about the assessment.

4.2.1 Perform a schema assessment

1. Navigate to *Using Migration Portal*.
2. Enter your EDB credentials.
3. Select **New** to create a project.
4. In the dialog box, enter the project name in the **Projectname** field.
5. Specify project details:
 - Select the application interface. - Select a source DB and version for the source. - Select the target DB and Version for the target.
6. To upload the *.SQL* file generated with your schema metadata, select **Choosefile**.

A screenshot of a 'New project' dialog box. The dialog has a title bar with a close button. It contains several sections: 'Project name' with a text input field containing 'e.g. First project'; 'Application interface' with radio buttons for 'JDBC' (selected), 'ODBC', '.NET', 'OCI', 'ProC', and 'Other'; 'Source DB' and 'Version' dropdowns with 'Oracle' and '11g' selected; 'Target DB' and 'Version' dropdowns with 'EDB Postgres Advance...' and '13' selected; 'DDL file' with a text input showing 'No file chosen' and a 'Choose file' button; a checkbox for 'Add suffix to make object names unique' with an info icon; and a 'Description(Optional)' text area with the placeholder 'Say something about the project...' and a '200 characters left' indicator. At the bottom are 'Cancel' and 'Create & assess' buttons.

New project

Project name
e.g. First project

Application interface
☒ JDBC ☐ ODBC ☐ .NET ☐ OCI ☐ ProC ☐ Other

Source DB
Oracle

Version
11g

Target DB
EDB Postgres Advance...

Version
13

DDL file [\(How to get a DDL file?\)](#)
No file chosen [Choose file](#)

☐ Add suffix to make object names unique ⓘ

Description(Optional)
Say something about the project...
200 characters left

[Cancel](#) [Create & assess](#)

Fig. 1: The Migration Portal New project dialog

Note: Don't modify the .SQL file.

7. To specify adding a suffix to indexes (_IDX) and constraints (_CON) when creating a project, select the **Addsuffix to make object names unique** check box. This selection ensures better assessment results, as EDB Postgres Advanced Server doesn't support the same name for tables and indexes whereas Oracle does.
8. Select **Create & assess** to create a new project and to assess the schemas.

The analysis tool reviews every construct and executes repair actions to improve compatibility with EDB Postgres Advanced Server. It flags any remaining errors that require manual intervention.

9. Verify the DDL objects (e.g., TABLES) that don't show a 100% success ratio.
10. Select the objects that aren't compatible with EDB Postgres Advanced Server and view the details. At the bottom of the middle panel, you can view repair action details.

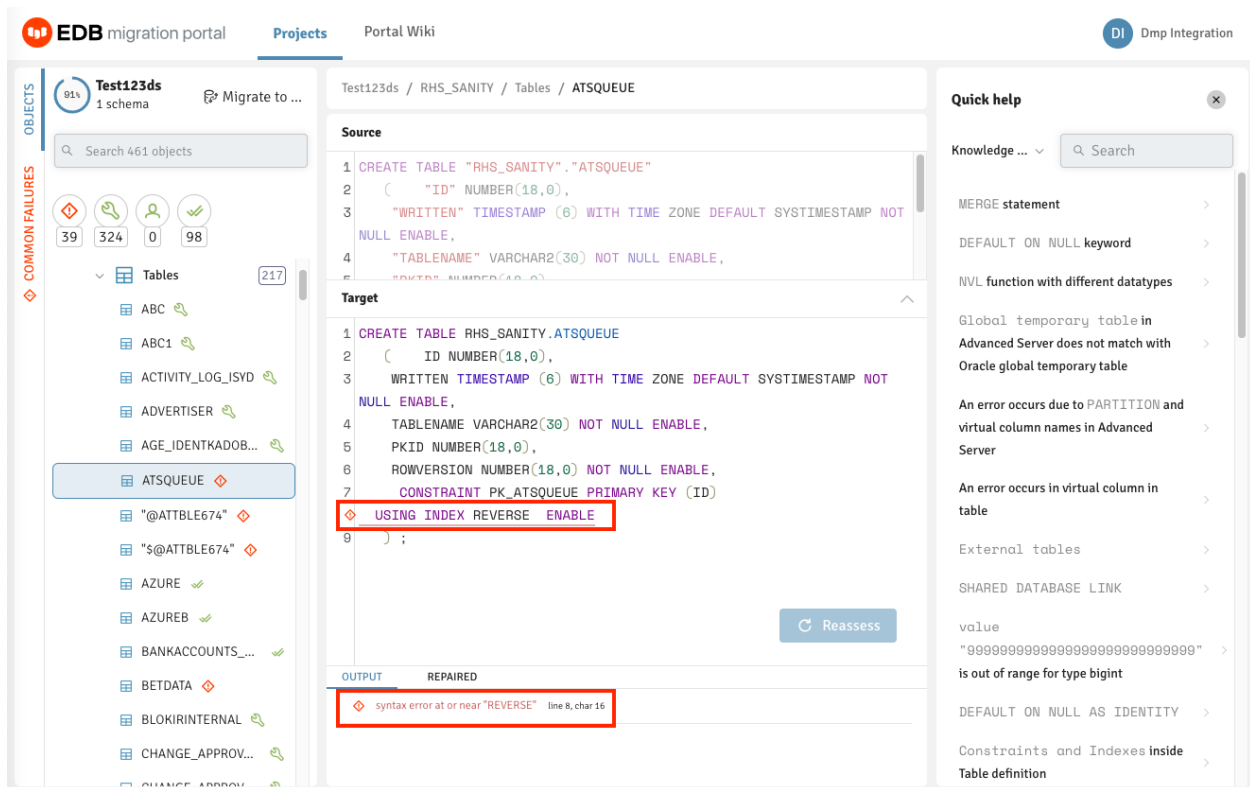


Fig. 2: Incompatible objects are identified

11. Refer to the Knowledge Base information in the right panel to locate the possible workarounds for the objects that aren't immediately compatible with EDB Postgres Advanced Server. You can also view the Knowledge Base information on the Portal wiki page.
12. On the **KnowledgeBase** tab, you can enter the error message for the incompatible objects with EDB Postgres Advanced Server and select **Search**.

The object detail panel displays the workaround or the resolution for the failed object. You can manually make the changes on the **Assessment** tab for that object and select **Reassess**.

Note: If any failed object passes while reassessing, the dependent objects for that object are also reassessed.

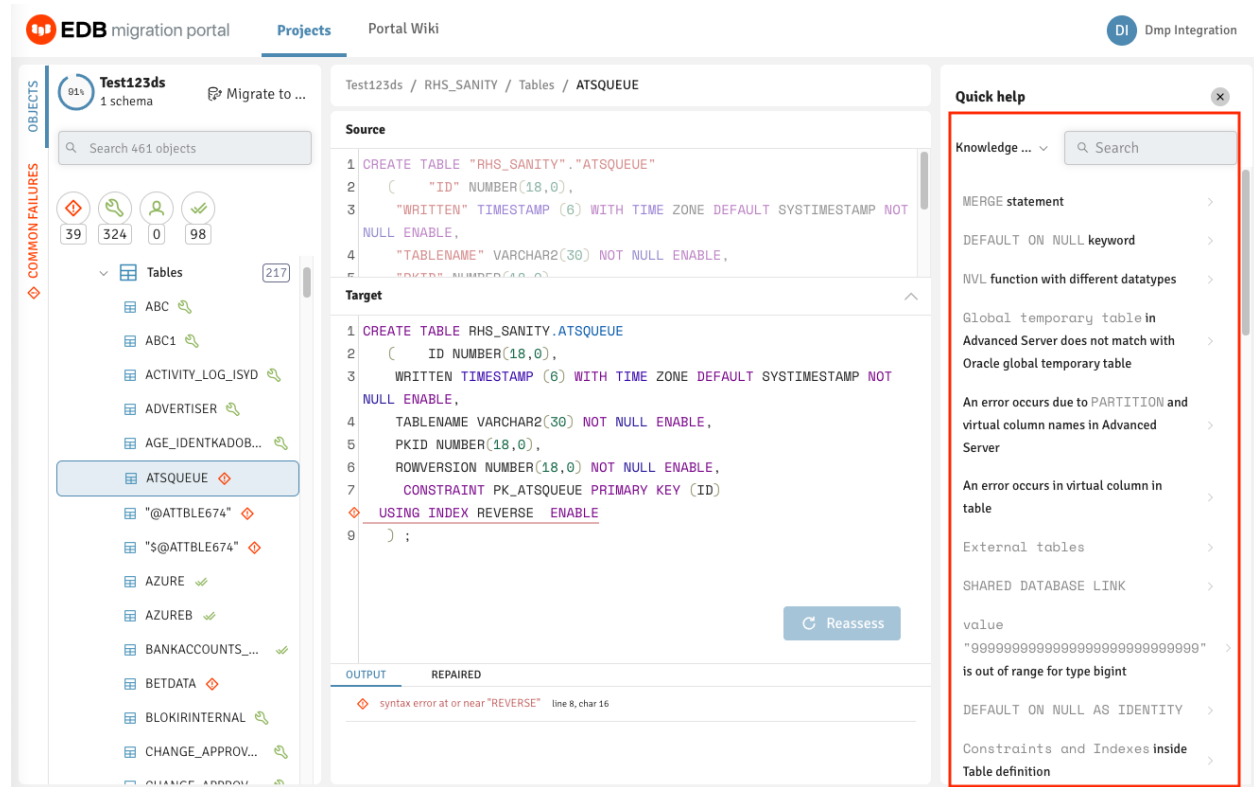


Fig. 3: Incompatible objects are identified

Similarly, you can make all incompatible objects compatible.

Note: If the information or workaround for incompatible objects isn't available in the Knowledge Base, contact the Support team for help.

When you have finished working with the DDL, you can either download the modified EDB-compatible DDL as a .sql file or migrate the schemas. See [Schema migration](#).

4.2.2 Generate an assessment report

Migration Portal reports provide a high-level assessment summary of the schemas assessed for your project. In addition, the report provides details about the failed objects and the cause of failure.

1. Select **Report**.
2. Select the schemas to include in the report.
3. To generate the onscreen report, select **Generate**.

You can optionally select **GeneratePDF** to save the report in PDF format. You can also view the count of distinct repair handlers applied to the DDLs under the selected schemas.

Note: Migration Portal doesn't assess sensitive PL/SQL source code hidden in Oracle wrapped objects. These wrapped objects aren't included in the assessed objects count, and therefore the true value of compatibility percentage might be

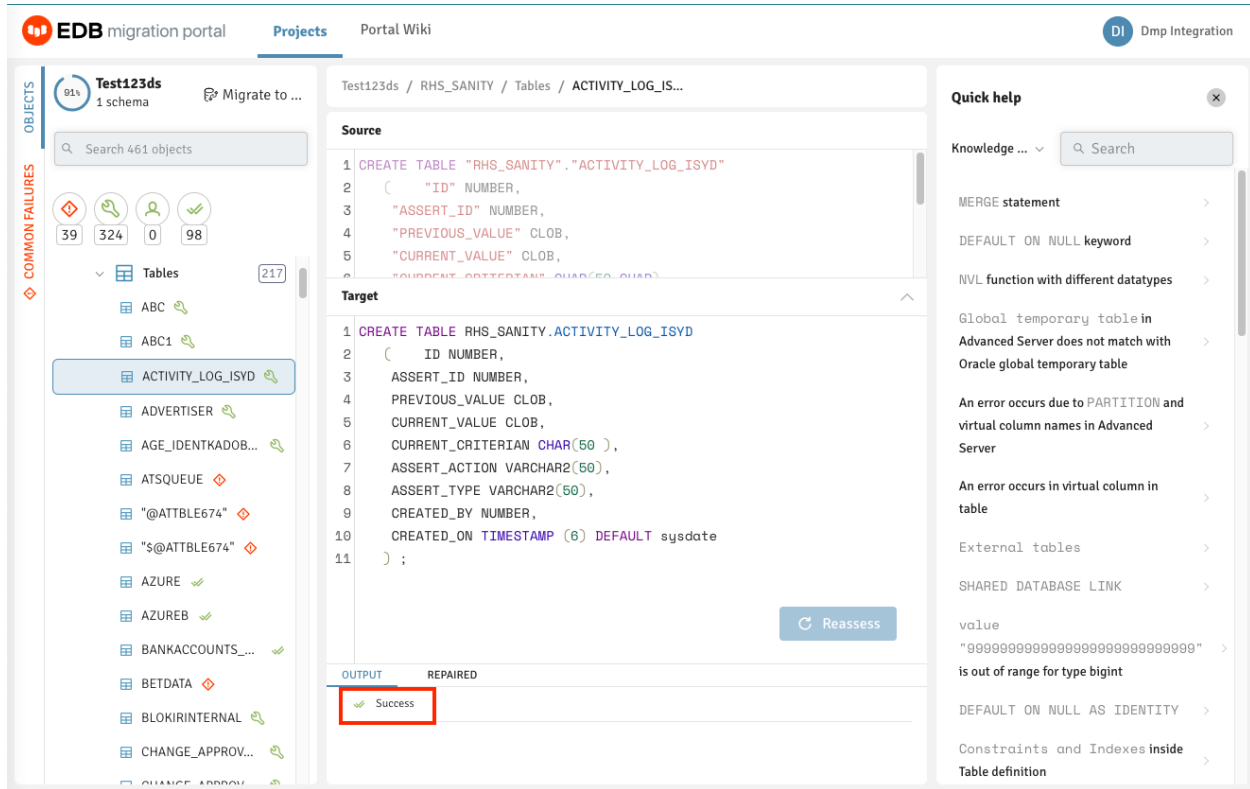


Fig. 4: Workaround or resolution for incompatible objects

different from the value calculated in the assessment report. See the note in [Performing a schema extraction](#) for more information about wrapped objects.

4.3 Schema migration

After resolving errors in your schemas, you can use the schemas with a client application such as pgAdmin, Toad Edge, or the PSQL client. Or you can migrate the schema to an EDB Postgres Advanced Server.

You can migrate to any of these schemas:

- Existing on-premises EDB Postgres Advanced Server
- New on-premises EDB Postgres Advanced Server
- EDB Postgres Advanced Server in the cloud

4.3.1 Migrating to an existing on-premises EDB Postgres Advanced Server host

You can migrate schemas to an existing on-premises EDB Postgres Advanced Server on Windows or Linux platforms.

Migrate schemas on Windows

1. Select **Existingon-premisesEDBPostgresPostgresAdvancedServer** .
2. Select one or more schemas to migrate to EDB Postgres Advanced Server.

Note: If your schemas aren't 100% compatible, a banner appears. Complete the Contact Us form as needed.

3. To download the assessed schemas, select **DownloadSQLfile** .
4. Select **Windows** .
5. To import the schemas, run:
 - On CLI:

```
\i c:\users\...\<project_name>.sql
```

- On cmd/shell:

```
edb-psql -f <project_name>.sql
```

Note: You can alternatively use the *pgAdmin* client for the import.

The converted schemas migrate to the target server.

Migrate schemas on Linux

You can migrate schemas to an existing on-premises EDB Postgres Advanced Server on Linux.

1. Select **Existingon-premisesEDBPostgresAdvancedServer** .
2. Select one or more schemas to migrate to EDB Postgres Advanced Server.
3. To download the assessed schemas, select **DownloadSQLfile** .
4. Select **Linux** .
5. To import the schemas, invoke the following `edb-psql` client commands:

```
sudo su - enterprisedb
edb-psql edb
create database <database_name>;
\connect <database_name>
\i <project_name>.sql
```

Note: You can alternatively use the *pgAdmin* client for the import.

The converted schemas migrate to the target server.

4.3.2 Migrating to a new on-premises EDB Postgres Advanced Server installation

You can install new EDB Postgres Advanced Server on premises on Windows or Linux platforms and migrate the schemas.

Migrate schemas on Windows

Migrate schemas to a new on-premises EDB Postgres Advanced Server on Windows.

1. Select **Newon-premisesEDBPostgresPostgresAdvancedServer** .
2. Select one or more schemas to migrate on EDB Postgres Advanced Server.
3. For the operating system, select **Windows** .
4. Select **DownloadWindowsinstaller** .
5. For installation steps, select **EDBPostgresAdvancedServerInstallationGuideforWindows** .
6. To download the assessed schemas, select **DownloadSQLfile** .
7. You can import schemas by running:
 - On CLI

```
\i c:\users\...\<project_name>.sql
```

- On cmd/shell

```
edb-psql -f <project_name>.sql
```

Note: You can alternatively use the *pgAdmin* client for the import.

Migrate schemas on Linux

Migrate schemas to an on-premises EDB Postgres Advanced Server on Linux.

1. Select **NewOn-premisesEDBPostgresAdvancedServer** .
2. Select one or more schemas to migrate on EDB Postgres Advanced Server.
3. For the operating system, select **Linux** .
4. You can select one of the following options to install the EDB Postgres Advanced Server:
 - Repository - More options
5. For information on the installation procedure, select **EDBPostgresAdvancedServerInstallationGuide** for Linux.
6. To download the assessed schemas, select **DownloadSQLfile** .
7. To import the schemas, run:

```
sudo su - enterprisedb
edb-psql edb
create database <database_name>;
\connect <database_name>
\i <project_name>.sql
```

Note: You can alternatively use the *pgAdmin* client for the import.

The converted schemas migrate to the target server.

4.3.3 Migrate to the cloud

Migrate schemas on EDB Postgres Advanced Server to the cloud.

1. Select **EDBPostgresAdvancedServeronCloud** .
2. Select one or more schemas to migrate to EDB Postgres Advanced Server.
3. Select the cloud platform, for example, **BigAnimal** .
4. To launch a new cluster, select **GotoBigAnimal** . Or, if you have an existing cluster running, select **Next** .

Note: See the [Big Animal](#) page for more information.

5. Enter the required connection details on the **Connect** page.
7. To verify the connection details, select **TestConnection** .

Note: You can select **Edit Connection** to make changes to the connection details and retest the connection details.

8. Once the connection is successful, select **Next** .

The converted schemas migrate to the target server.

4.4 Data migration

After performing the schema migration, migrate the data.

1. Use Migration Toolkit to migrate the data. For detailed information about using Migration Toolkit, see Migration Toolkit .
2. Configure the Migration Toolkit `toolkit.properties` file, ensuring that connection information for the source and target databases is available in the property file:

```
SRC_DB_URL = jdbc: oracle:thin:@localhost:1521:ORCL
SRC_DB_USER = user_name
SRC_DB_PASSWORD = password
TARGET_DB_URL= jdbc:edb://localhost:5444/migration
TARGET_DB_USER = enterprisedb
TARGET_DB-PASSWORD = password
```

For more information, see Building the toolkit.properties File .

3. Invoke Migration Toolkit in `-dataOnly` mode. Include the `-truncLoad` keyword to resolve foreign key dependencies across tables.

For example, the following command migrates data to a schema named `hr` :

```
runMTK.sh -dataOnly -targetSchema hr -truncLoad HR
```

Migrate your schemas to the cloud

1 Select Schemas

2 Select Platform

3 Launch Cluster

4 Connect

5 Deploy

6 Finish

Project Name

TestMigration

Interface

JDBC

Source DB

Oracle 11g

Target DB

EDB Postgres Advanced Server 13

Compatibility

99%

Connect to BigAnimal cluster

Target Database

TestMigration

Host Name/Address

Host Name/Address

Port

5432

Note: Ensure that your database server is accessible from Migration Portal.

Maintenance Database

edb_admin

Username

edb_admin

Password

Password

Test Connection

Cancel

Previous

Next

Fig. 5: Connecting to the cloud cluster

The command migrates the specified `source_schema` to `target_schema` . The data is loaded into the locally installed EDB Postgres Advanced Server instance with a database superuser named `enterprisedb` and the password of `password` .

Note: The tables are truncated before attempting the data load.

ADVANCED DATA MIGRATION

For larger databases that require a parallel data load, you can use one of the following methods:

- The EDB Postgres Advanced Server database link feature (for compatibility with Oracle databases)
- A Dblink or a database link style migration (if your data contains CLOB data)

For more information, see the Migration Toolkit .